R-IAC : Robust Intrinsically Motivated Active Learning

Adrien Baranes, Pierre-Yves Oudeyer

R-IAC : Robust Intrinsically Motivated Active Learning

Adrien Baranes and Pierre-Yves Oudeyer
INRIA Bordeaux Sud-Ouest – FLOWERS team

***Abstract*— IAC was initially introduced as a developmental mechanisms allowing a robot to self-organize developmental trajectories of increasing complexity without pre-programming the particular developmental stages. In this paper, we argue that IAC and other intrinsically motivated learning heuristics could be viewed as active learning algorithms that are particularly suited for learning forward models in unprepared sensorimotor spaces with large unlearnable subspaces. Then, we introduce a novel formulation of IAC, called R-IAC, and show that its performances as an intrinsically motivated active learning algorithm are far superior to IAC in a complex sensorimotor space where only a small subspace is neither unlearnable nor trivial. We also show results in which the learnt forward model is reused in a control scheme.**

***Index Terms*— active learning, intrinsically motivated learning, developmental robotics, artificial curiosity, sensorimotor learning.**

I. Intrinsically Motivated Exploration and Learning

Developmental robotics approaches are studying mechanisms that may allow a robot to continuously discover and learn new skills in unknown environments and in a life-long time scale [1], [2]. A main aspect is the fact that the set of these skills and their functions are at least partially unknown to the engineer who conceive the robot initially, and are also task-independent. Indeed, a desirable feature is that robots should be capable of exploring and developing various kinds of skills that they may re-use later on for tasks that they did not foresee. This is what happens in human children, and this is also why developmental robotics shall import concepts and mechanisms from human developmental psychology.

A. Learn from the Real Experimentations

Like children, the "freedom" that is given to developmental robots to learn an open set of skills also poses a very important problem: as soon as the set of motors and sensors is rich enough, the set of potential skills become extremely large and complicated. This means that on the one hand, it is impossible to try to learn all skills that may potentially be learnt because there is not enough time, and also that there are many skills or goals that the child/robot could imagine but never be actually learnable, because they are either too difficult or just not possible (for example, trying to learn to control the weather by producing gestures is hopeless). This kind of problem is not at all typical of the existing work in machine learning, where usually the "space" and the associated "skills" to be learnt and explored are well-prepared by a human engineer. For example, when learning hand-eye coordination in robots, the right input and output spaces (e.g. arm joint parameters and visual position of the hand) are typically provided as well as the fact that hand-eye coordination is an interesting skill to learn. But a developmental robot is not supposed to be provided with the right subspaces of its rich sensorimotor space and with their association with appropriate skills: it would for example have to discover that arm joint parameters and visual position of the hand are related in the context of a certain skill (which we call hand-eye coordination but which it has to conceptualize by itself) and in the middle of a complex flow of values in a richer set of sensations and actions.

B. Intrinsic motivations

Developmental robots have a sharp need for mechanisms that may drive and self-organize the exploration of new skills, as well as identify and organize useful sub-spaces in its complex sensorimotor experiences. In psychology terms, this amount to trying to answer the question "What is interesting for a curious brain?". Among the various trends of research which have approached this question, of particular interest is work on intrinsic motivation. Intrinsic motivations are mechanisms that guide curiosity-driven exploration, that were initially studied in psychology [3]-[5] and are now also being approached in neuroscience [6]-|8]. Machine learning researchers have proposed that such mechanism might be crucial for self-organizing developmental trajectories as well as for guiding the learning of general and reusable skills in machines and robots [9,10]. Experiments have been conducted in real-world robotic setups, such as in [9] where an intrinsic motivation system was shown to allow for the progressive discovery of skills of increasing complexity, such as reaching, biting and simple vocal imitation with and AIBO robot. In these experiments, the focus was on the study of how developmental stages could self-organize into a developmental trajectory without a direct pre-specification of these stages and their number.

II. Robust Intelligent Adaptive Curiosity (**RIAC**) as active learning

The present paper aims to propose a new version of the algorithm called Intelligent Adaptive Curiosity (**IAC**) presented in [10], and to show that it can be used as an efficient active learning algorithm to learn forward models in a complex unprepared sensorimotor space. This algorithm, based on intrinsic motivations heuristics, implements an active and adaptive mechanism for monitoring and controlling the growth of complexity in exploration and incremental learning.

In [9], it was presented focusing on its ability to generate organized developmental stages and trajectories within a cognitive modeling endeavour. Here, we rather take an engineering approach to study how IAC and a new formulation, called **Robust-IAC (R-IAC)**, can efficiently drive the robot to learn fast and correctly a forward model.

A. Developmental Active Learning

An essential activity of epigenetic robots is to learn forward models of the world, which boils down to learning to predict the consequences of its actions in given contexts. This learning happens as the robot collects learning examples from its experiences. If the process of example collection is disconnected from the learning mechanism, this is called passive learning. In contrast, researchers in machine learning have proposed algorithms allowing the machine to choose and make experiments that maximize the expected information gain of the associated learning example [11], which is called "active learning". This has been shown to dramatically decrease the number of required learning examples in order to reach a given performance in data mining experiments [12], which is essential for a robot since physical action costs time and energy. We argue that intrinsically motivated learning with **IAC** can be considered as an "active learning" algorithm. We will show that some of them allow very efficient learning in unprepared spaces with the typical properties of those encountered by developmental robots, outperforming standard active learning heuristics.

The typical active learning heuristics consist in focusing the exploration in zones where unpredictability or uncertainty of the current internal model are maximal, which involves the online learning of a meta-model that evaluates this unpredictability or uncertainty.

Unfortunately, it is not difficult to see that it will fail completely in unprepared robot sensorimotor spaces. Indeed, the spaces that epigenetic robots have to explore are typically composed of unlearnable subspaces, such as for example the relation between its joints values and the motion of unrelated objects that might be visually perceived. Classic active learning heuristics will push the robot to concentrate on these unlearnable zones, which is obviously undesirable.

Based on psychological theories proposing that exploration is focused on zones of optimal intermediate difficulty or novelty [13], [14], intrinsic motivation mechanisms have been proposed, pushing robots to focus on zones of maximal learning progress [9]. As exploration is here closely coupled with learning, this can be considered as active learning. We will now present the IAC system together with its novel formulation R-IAC. After this, we will evaluate their active learning performances in an inhomogeneous sensorimotor space with unlearnable subspaces.

B. Prediction Machine and Analysis of Error Rate

We consider a robot as a system with motor channels **M** and sensori channels **S (M** and **S** can be low-level such as torque motor values or touch sensor values, or higher level such as a "go forward one meter" motor command or "face detected" visual sensor"). Real valued action/motor parameters are represented as a vector $\mathbf{M(t)}$, and sensors, as $\mathbf{S(t)}$, at a time t. $\mathbf{SM(t)}$ represents a sensorimotor context, i.e. the concatenation of both motors and sensors vectors.

We also consider a Prediction Machine **PM,** as a system based on a learning algorithm (neural networks, KNN, etc.), which is able to create a forward model of a sensorimotor space based on learning examples collected through self-experiments. Experiments are defined as series of actions, and consideration of sensations detected after actions are performed. An experiment is represented by the set $(\mathbf{SM(t)}, \mathbf{S(t+1)})$**,** and denotes the sensori consequence **S(t+1)** that is observed when actions encoded in **M(t)** are performed in the sensori context **S(t).** This set is called a "**learning exemplar**". After each trial, the prediction machine **PM** gets this data and incrementally updates the forward model that it is encoding, i.e. the robot incrementally increases its knowledge of the sensorimotor space. In this update process, **PM** is able to compare, for a given context $\mathbf{SM}(\boldsymbol{t})$, differences between predicted sensations $\tilde{\mathbf{S}}(\boldsymbol{t+1})$ (estimated using the created model), and real consequences $\mathbf{S}(\boldsymbol{t+1})$. It is then able to produce a measure of error $\boldsymbol{e(t+1)}$**,** which represents the quality of the model for ensorimotor context $\mathbf{SM}(\boldsymbol{t})$.

Then, we consider a module able to analyze learning evolutions over time, called Prediction Analysis Machine **PAM**, Fig. 1. In a given subregion $\boldsymbol{R}$ of the sensorimotor space, this system monitors the evolution of errors in predictions made my **PM** by computing its derivative, i.e. the learning progress, $\mathbf{LP}=\boldsymbol{e_N}-\boldsymbol{e_F}$ in this particular region over a sliding time window (see Fig 1). **LP** is then used as a measure of interestingness used in the action selection scheme outlined below. The more a region is characterized by learning progress, the more it is interesting, and the more the system will perform experiments and collect learning examplars that fall into this region. Of course, as exploration goes on, the learnt forward model becomes better in this region and learning progress might decrease, leading to a decrease in the interestingness of this region.

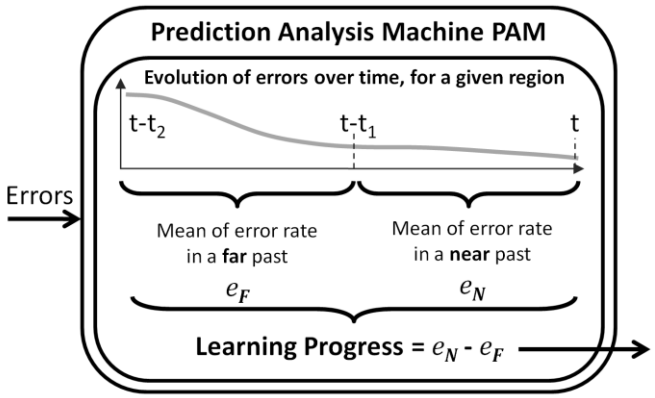

Fig. 1. Internal mechanism of the Prediction Analysis Machine **PAM** associated to a given subregion **R** of the sensorimotor space. This module considers errors detected in prediction by the Prediction Machine **PM**, and returns a value representative of the learning progress in the region. Learning progress is the derivative of errors analyzed between a far and a near past in a fixed length sliding window.

To precisely represent the learning behavior inside the whole sensorimotor space and differentiate its various evolutions in various subspaces/subregions, different **PAM** modules, each associated to a different subregion R_i of the sensorimotor space, need to be built. Therefore, the learning progress provided as the output values of each **PAM** (called D_i the Fig. 2.) become representative of the interestingness of the associated region R_i. Initially, the whole space is considered as one single region R_0, associated to one **PAM**, which will be progressively split into subregions with their own **PAM** as we will now describe.

C. The Split Machine

The Split Machine **SpM** possesses the capacity to memorize all the experimented learning exemplars $(\mathbf{SM(t)}, \mathbf{S(t+1)})$, and the corresponding errors values $\boldsymbol{e(t+1)}$. It is both responsible for identifying the region and **PAM** corresponding to a given **SM(t)**, but also responsible of splitting (or creating in R-IAC where parent regions are kept in use) sub-regions from existing regions.

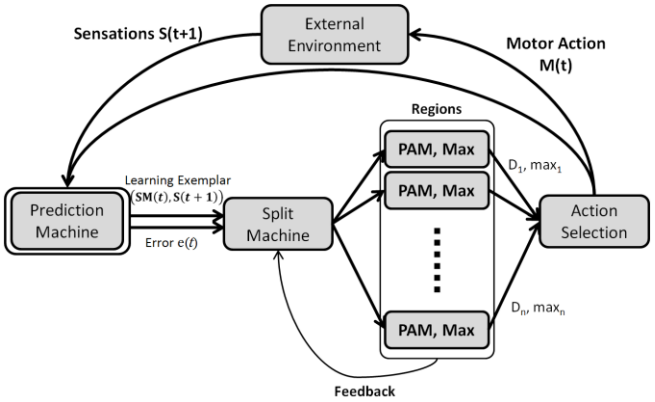

Fig. 2. General architecture of IAC and R-IAC. The prediction Machine is used to create a forward model of the world, and measures the quality of its predictions (errors values). Then, a split machine cuts the sensorimotor space into different regions, whose quality of learning over time is examined by Prediction Analysis Machines. Then, an Action Selection system, is used to choose experiments to perform.

1) Region Implementation

We use a tree representation to store the list of regions as shown in Fig. 3. The main node represents the whole space, and leafs are subspaces. $\mathbf{S(t)}$ and $\mathbf{M(t)}$ are here normalized into $[0;1]^n$. The main region (first node), called R_0, represents the whole sensorimotor space. Each region stores all collected examplars that it covers. When a region contains more than a fixed number $\mathbf{T_{split}}$ of exemplars, we split it into two ones.

When this criterion has been reached by a region, the split algorithm is executed, splitting just in one dimension at a time. An example of split execution is shown in Fig. 3, using a two dimensions input space.

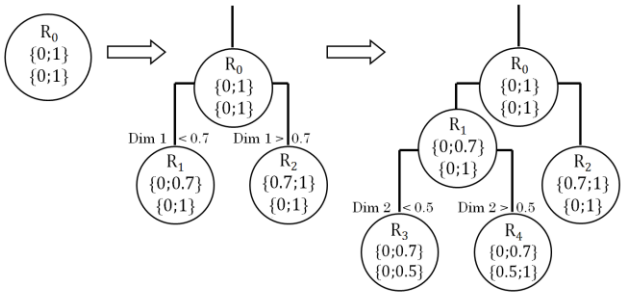

Fig. 3. The sensorimotor space is iteratively and recursively split into sub-spaces, called "regions". Each region R_n is responsible for monitoring the evolution of the error rate in the anticipation of consequences of the robot's actions, if the associated contexts are covered by this region.

2) *IAC* Split Algorithm

In the **IAC** algorithm, the idea was to find a split such that the two sets of exemplars into the two subregions would minimize the sum of the variances of $\mathbf{S(t+1)}$ components of exemplars of each set, weighted by the number of exemplars of each set. The idea was to split in the middle of zones of maximal change in the function **SM(t) → S(t+1)**. Mathematically, we consider $\varphi_n = \left\{ \left(\mathbf{SM}(\boldsymbol{t}), \mathbf{S}(\boldsymbol{t+1})\right)_i \right\}$ as the set of exemplars possessed by region R_n. Let us denote j a cutting dimension and v_j, an associated cutting value. Then, the split of φ_n into φ_{n+1} and φ_{n+2} is done by choosing j and v_j such that:

(1) All the exemplars $(\mathbf{SM}(\boldsymbol{t}), \mathbf{S}(\boldsymbol{t+1}))_i$ of φ_{n+1} have a j^{th} component of their $\mathbf{SM}(\boldsymbol{t})$ smaller than v_j

(2) All the exemplars $(\mathbf{SM}(\boldsymbol{t}), \mathbf{S}(\boldsymbol{t+1}))_i$ of φ_{n+2} have a j^{th} component of their $\mathbf{SM}(\boldsymbol{t})$ greater than v_j

(3) The quantity :

$$|\varphi_{n+1}|.\sigma\left(\left\{\mathbf{S}(\boldsymbol{t+1})|\left(\mathbf{SM}(\boldsymbol{t}), \mathbf{S}(\boldsymbol{t+1})\right) \in \varphi_{n+1}\right\}\right) + |\varphi_{n+2}|.\sigma\left(\left\{\mathbf{S}(\boldsymbol{t+1})|\left(\mathbf{SM}(\boldsymbol{t}), \mathbf{S}(\boldsymbol{t+1})\right) \in \varphi_{n+2}\right\}\right)$$

is **minimal**, where

$$\sigma(\mathrm{S}) = \frac{\sum_{v \in \mathrm{S}} \left\| s - \frac{\sum_{v \in S} v}{|\mathrm{S}|} \right\|^2}{|\mathrm{S}|}$$

where S is a set of vectors, and |S|, its cardinal.

3) ***R-IAC*** *Split Algorithm*

In **R-IAC**, the splitting mechanism is based on comparisons between the learning progress in the two potential child regions. The principal idea is to perform the **separation which maximizes the dissimilarity of learning progress** comparing the two created regions. This leads to the direct detection of areas where the learning progress is maximal, and to separate them from others (see Fig. 4). This contrasts with **IAC** where regions where built independently of the notion of learning progress.

Reusing the notations of the previous section, in **R-IAC** the split of φ_n into φ_{n+1} and φ_{n+2} is done by choosing j and v_j such that:

$$\left(LP\left(\left\{\mathbf{e}(\boldsymbol{t+1})|\left(\mathbf{SM}(\boldsymbol{t}), \mathbf{S}(\boldsymbol{t+1})\right) \in \varphi_{n+1}\right\}\right) - LP\left(\left\{\mathbf{e}(\boldsymbol{t+1})|\left(\mathbf{SM}(\boldsymbol{t}), \mathbf{S}(\boldsymbol{t+1})\right) \in \varphi_{n+2}\right\}\right)\right)^2$$

is **maximal**, where

$$LP(E) = \frac{\sum_{i=1}^{\frac{|E|}{2}} e(i) - \sum_{i=\frac{|E|}{2}}^{|E|} e(i)}{|E|}$$

where E is a set of errors values $\{e(i)\}$ with errors indexed by their relative order i of encounter (e.g. error *e(9)* corresponds to a prediction made by the robot before another prediction which resulted in *e(10)*: this implies that the order of examplars collected and associated prediction errors are stored in the system). $|E|$ is the cardinal of this set, and $LP(E)$ is the learning progress responsible of the computation of learning progress estimations.

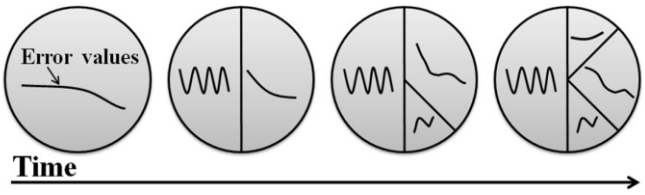

Fig. 4. Evolution of the sensorimotor regions over time. The whole space is progressively subdivided in such a way that the dissimilarity of each sub-region in terms of learning progress is maximal.

D. Action Selection Machine

We present here an implementation of Action Selection Machine **ASM**. The **ASM** decides of actions $\mathbf{M}(\boldsymbol{t})$ to perform, given a sensori context $\mathbf{S}(\boldsymbol{t})$. (See Fig. 2.). The ASM heuristics is based on a mixture of several **modes**, which differ between **IAC** and **R-IAC**. Both **IAC** and **R-IAC** algorithms share the same global loop in which modes are chosen probabilistically:

Loop:

- **Action Selection Machine ASM**: given **S(t)**, execute an action $\mathbf{M}(\boldsymbol{t})$ using the **mode** $(\boldsymbol{n})$ with probability $\boldsymbol{p_n}$and based on data stored in the region tree;
- **Prediction Machine PM**: Estimate the predicted consequence $\tilde{\boldsymbol{S}}_{\boldsymbol{t+1}}$ using the prediction machine **PM ;**
- **External Environment**: Measure the real consequence $\boldsymbol{S}_{\boldsymbol{t+1}}$
- **Prediction Machine PM**: Compute the error $\boldsymbol{e(t+1)} = \boldsymbol{abs}(\tilde{\boldsymbol{S}}_{\boldsymbol{t+1}} - \boldsymbol{S}_{\boldsymbol{t+1}})$**;**
- Update the **prediction machine PM** with $(\mathbf{SM(t)}, \mathbf{S(t+1)})$
- **Split Machine SpM:** update the region tree with $(\mathbf{SM(t)}, \mathbf{S(t+1)})$ and $\boldsymbol{e(t+1)}$**;**
- **Prediction Analysis Machine PAM:** update evaluation of learning progress in the regions that cover $(\mathbf{SM(t)}, \mathbf{S(t+1)})$

End Loop

We now present the different exploration modes used by the Action Selection Machine, in **IAC** and **R-IAC** algorithm:

1) Random Babbling Exploration Mode **(1)**

The **random babbling** mode corresponds to a totally random exploration, which does not consider previous actions and context. This mode appears in both **IAC** and **R-IAC** algorithm, with a probability $\boldsymbol{p_1} = 30\%$**.**

2) Learning Progress Maximization Exploration Mode **(2)**

The learning progress maximization mode aims to maximize the informational gain obtained after each experiment. To do this, it considers data computed by all **PAM**. The main idea is to consider that regions which have maximum learning progress are potentially the more interesting to explore. In the **IAC** algorithm, mode *2* action selection is straightforward: the leaf region which learning progress is maximal is found, and a random action within this region is chosen. In the **R-IAC** algorithm, we take into account the fact that many regions may have close learning progress values by taking a probabilistic approach. Furthermore, instead of focusing on the leaf regions like in **IAC**, **R-IAC** continues to monitor learning progress in node regions and select them if they have more learning progress. Let us give more details:

a) Probabilistic approach

The probabilistic mechanism is based on the consideration of a probability to choose a region proportional to the learning progress. We have, for a set of derivatives $D = \{D_0, D_1 \dots D_n\}$ representing each region $\{R_1, R_2 \dots R_n\}$ using **PAM**, the probabilities $\boldsymbol{P_n}$to choose the region R_n as :

$$\boldsymbol{P_n} = \frac{|D_n - max(D_i)|}{\sum_{\boldsymbol{i}} |D_i - max(D_i)|}$$

b) Multiresolution Monitoring of Learning Progress

In the **IAC** algorithm, the estimation of learning progress only happens in leaf regions. In **R-IAC**, learning progress is monitored in all regions created during the system's life time, which allows to track learning progress at multiple resolution in the sensorimotor space. This implies that when a new exemplar is available, **R-IAC** updates the evaluation of learning progress in all regions that cover this exemplar (but only if the exemplar was chosen randomly, i.e. not with mode *3* as described below).

In **R-IAC mode *2***, when a region has been chosen with the probabilistic approach and mutiresolution scheme, a random action is chosen within this region. Mode 2 is typically chosen with a probability $\boldsymbol{p_2} = 60\%$ in both **R-IAC** and $\boldsymbol{p_2} = 70\%$ in **IAC** (which means this is the dominant mode).

3) Error Maximization Exploration Mode **(3)**

Mode 3 combines a traditional active learning heuristics with the concept of learning progress: in mode 3, a region is first chosen with the same scheme as in **R-IAC** mode 2. But once this region has been chosen, an action in this region is selected such that the expected error in prediction will be maximal. This is currently implemented through a k-nearest neighbor regression of the function SM(t) → e(t+1) which allows to find the point of maximal error, to which is added random noise (to avoid to query several times exactly the same point). Like shown in Fig. 2, we store, for each region, coordinates of the generated point (called "**MAX**"). Mode 3 is typically chose with a probability $\boldsymbol{p_3} = 60\%$ in R-IAC (and does not appear in **IAC**).

E. Consequences of Learning Progress Examination

The examination of learning progress allows a control of the learning complexity. Let us imagine three typical situations to illustrate precisely this phenomenon:

- The system is exploring a simple area: The learning rate is high during a brief instant, and then, it is approximately null. The derivative is thus constant, and the probability to explore this kind of area is low.
- The system is exploring a difficult area:The learning rate is varying very rapidly. The derivative is thus about zero, and the probability is low, like in the previous case.
- The system is exploring a zone of mean difficulty: The learning rate is increasing, the derivative is thus negative, and the probability is depending on its absolute value.

Observing these three examples, representing possible situations encountered, we argue that the learning progress is a guide toward areas of intermediate difficulty.

III. THE HAND-EYE-CLOUDS EXPERIMENT

We will now compare the performances of IAC and R-IAC as active learning algorithms to learn a forward model in a complex 6-dimensional sensorimotor space that includes large unlearnable zones. Both algorithms will also be compared with baseline random exploration.

In this experiment, a simulated robot has two 2-D arms with two joints controlled by motor **inputs** $\boldsymbol{q_{11}}$, $\boldsymbol{q_{12}}$, $\boldsymbol{q_{21}}$, $\boldsymbol{q_{22}}$. On the tip of one of the two arms is attached a square camera capable to detect the sensori position $(\boldsymbol{x}, \boldsymbol{y})$ of point-blobs (relative to the square). These point-blobs can be either the tip of the other arm or clouds in the sky (see figure 5). This means that when the right arm is positioned such that the camera is over the clouds, which move randomly, the relation between motor configurations and perception is quasi-random. If on the contrary the arms are such that the camera is on top of the tip of the other arm, then there is an interesting sensorimotor relationship to learn. Formally, the system has the relation:

$$(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{E}(\boldsymbol{q_{11}}, \boldsymbol{q_{12}}, \boldsymbol{q_{21}}, \boldsymbol{q_{22}})$$

where $(\boldsymbol{x}, \boldsymbol{y})$ is computed as follows:

(1) Nothing has been detected : the camera has been placed over the white wall, $(\boldsymbol{x}, \boldsymbol{y})$ = (-10, -10);

(2) The hand appears inside the camera: The value of the relative position of the hand in the camera referential $\boldsymbol{C}$ is taken. According to the camera size, the **x** and **y** values are in the interval [0; 6];

(3) The camera is looking at the window: Two random values playing the role of random clouds displacement are chosen for output. The interval of outputs corresponds to camera size.

(4) The camera is looking at the window and sees both hand and cloud: we choose a random output value, like if just a cloud had been detected.

This setup can be thought to be similar to the problems encountered by infants discovering their body: they do not know initially that among the blobs moving in their field of view, some of them are part of their "self" and can be controlled, such as the hand, and some other are independent of the self and cannot be controlled (e.g. cars passing in the street or clouds in the sky).

Thus, in this sensorimotor space, the "interesting" potentially learnable subsuspace is next to a large unlearnable subspace (unlearnable), and also next to a large very simple subspace (when the camera is looking neither the clouds not the tip of the other arm). The primary challenge is thus to avoid the noisy area, and to detect others.

Results. In these experiments, the parameters of **IAC** and **R-IAC** are Tsplit=250, the learning progress window is 50, $\boldsymbol{p_1}$ = 0.3, $\boldsymbol{p_2} = 0.6$, $\boldsymbol{p_3} = 0.1$. Experiments span a duration of 100000 actions. The learning algorithm that is used to learn the forward model is an incremental version of Gaussian Mixture Regression (Calinon et al., 2007).

A first study of what happens consists in monitoring the distance between the center of the eye (camera), and the hand (tip of the other arm). A small distance means that the eye is looking the hand, and a high, that it is focusing on clouds (noisy part) or on the white wall. Fig. 6 shows histograms of these distances. We firstly observe the behavior of the Random exploration algorithm. The curve shows that the system is, in majority, describing actions with a distance of 22, corresponding to the camera looking at clouds or at the white wall. Interestingly, the curve of the **IAC** algorithm is similar but slightly displaced towards shorter distance: this shows that **IAC** pushed the system to explore the "interesting" zone a little more. We finally observe that **RIAC** shows a large difference with both IAC and Random exploration: the system spends three times more time in a distance inferior to 8, i.e. exploring sensorimotor configurations in which the camera is looking at the other arm's tip. Thus, the difference between **R-IAC** and **IAC** is more important than the difference between **IAC** and Random exploration.

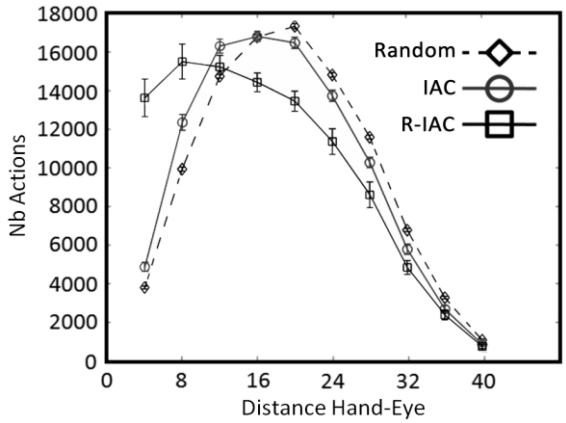

Fig. 6. Histogram of distances repartitions between hand and eye, after 100000 experiments, comparing **Random**, **IAC** and **R-IAC** exploration methods.

Then, we evaluated the quality of the forward model learnt using the three exploration algorithms. We considered this quality in two respects: 1) the capability of the model to predict the position of the hand in the camera given motor configurations for which the hand is within the field of view of the robot; 2) the capacity to use the forward model to control the arm: given a right arm configuration and a visual

objective, we tested how far the forward model could be used to drive the hand to reach this visual objective.The first kind of evaluation was realized by first building a test database of 1000 random motor configurations for which the hand is within the field of view, and then using it for testing the learnt models built by each algorithm at various stages of their lifetime (the test consisted in predicting the position of the hand in the camera given joint confiurations). Results are reported on the right of figure 7. The second evaluation consisted in generating a set of $\{(\boldsymbol{x},\boldsymbol{y})_{\boldsymbol{C}}, \boldsymbol{q_{21}}, \boldsymbol{q_{22}}\}$ values that are possible given the morphology of the robot, and then use the learnt forward models to try to move the left arm to reach the $(\boldsymbol{x},\boldsymbol{y})_{\boldsymbol{C}}$ objectives corresponding to particular $\boldsymbol{q_{21}}, \boldsymbol{q_{22}}$ values**.** The distance between the reached point and the objective point was each time measured, and results are reported in the left graph of figure 7.

Both curves on figure 7 confirm clearly the qualitative results of the previous figure: **R-IAC** outperforms significantly **IAC**, which is only slighlty better than random exploration. We have thus shown that **R-IAC** is much more efficient in such an example of complex inhomogeneous sensorimotor space.

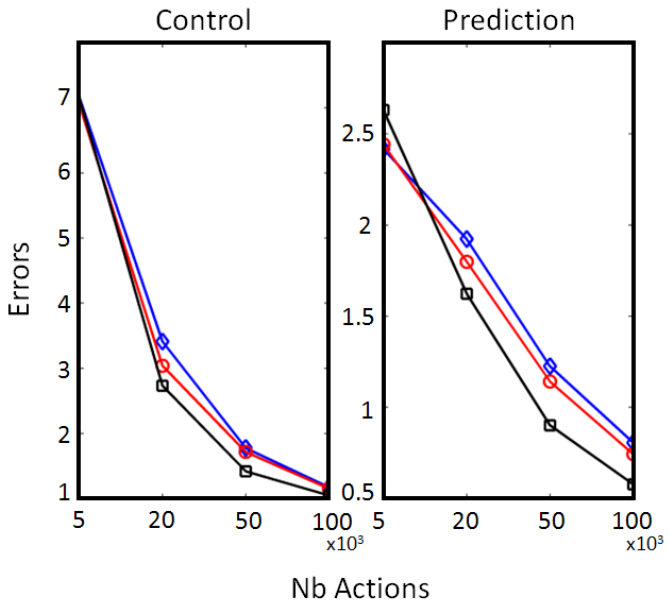

Figure 7 Left: evolution of performances in control based on the model learnt by Random exploration (blue line), **IAC** exploration (red line) and **R-IAC** exploration (black line). Right : evolution of the generalization capabilities in prediction of the learnt forward models with Random exploration (blue), **IAC** (red), and **R-IAC** (black)

IV. Conclusion

IAC was initially introduced as a developmental mechanism allowing a robot to self-organize developmental trajectories of increasing complexity without pre-programming the particular developmental stages. In this paper, we have argued that IAC and other intrinsically motivated learning heuristics could be viewed as active learning algorithms, and were based on heuristics that are more suited than traditional active learning algorithms for operation in unprepared sensorimotor spaces with large unlearnable subspaces. Then, we have introduced a novel formulation of **IAC**, called **R-IAC**, and shown that its performances as an intrinsically motivated active learning algorithm were far superior to **IAC** in a complex sensorimotor space where only a small subspace was interesting. We have also shown results in which the learnt forward model was reused in a control scheme.

V. References

[1] Weng, J., McClelland J., Pentland, A., Sporns, O., Stockman, I., Sur M., and Thelen, E. (2001) Autonomous mental development by robots and animals, Science, vol. 291, pp. 599–600.

[2] Lungarella, M., Metta, G., Pfeifer, R., and Sandini, G. (2003) Developmental robotics: A survey, Connection Sci., vol. 15, no. 4, pp. 151–190, 2003.

[3] White, R. (1959). Motivation reconsidered: The concept of competence. Psychological review, 66:297–333.

[4] Berlyne, D. (1966). Curiosity and Exploration, *Science,* Vol. 153. no. 3731, pp. 25 - 33

[5] Deci, E. and Ryan, R. (1985). Intrinsic Motivation and Self-Determination in Human Behavior. Plenum Press.

[6] Schultz, W. (2002) Getting Formal with Dopamine and Reward, Neuron, Vol. 36, pp. 241-263.

[7] Dayan, P. and Balleine, B. (2002) Reward, Motivation and Reinforcement Learning, Neuron, Vol. 36, pp. 285-298.

[8] Redgrave, P. and Gurney, K. (2006) The Short-Latency Dopamine Signal: a Role in Discovering Novel Actions?, Nature Reviews Neuroscience, Vol. 7, no. 12, pp. 967-975.

[9] Oudeyer P-Y & Kaplan , F. & Hafner, V. (2007) Intrinsic Motivation Systems for Autonomous Mental Development, IEEE Transactions on Evolutionary Computation, 11(2), pp. 265--286.

[10] Barto, A., Singh S., and Chentanez N. (2004) Intrinsically motivated learning of hierarchical collections of skills, in Proc. 3rd Int. Conf. Development Learn., San Diego, CA, 2004, pp. 112–119.

[11] Cohn D., Ghahramani Z., and Jordan M. (1996) Active learning with statistical models, J. Artif. Intell. Res., vol. 4, pp. 129–145, 1996.

[12] Hasenjager M.and Ritter H. (2002) Active Learning in Neural Networks. Berlin, Germany: Physica-Verlag GmbH, Physica-Verlag Studies In Fuzziness and Soft Computing Series, pp. 137–169.

[13] Berlyne, D. (1960). Conflict, Arousal, and Curiosity. New York: McGraw-Hill.

[14] Csikszentmihalyi, M. Creativity-Flow and the Psychology of Discovery and Invention. New York: Harper Perennial, 1996.

[15] Calinon, S., Guenter, F. and Billard, A. (2007). On Learning, Representing and Generalizing a Task in a Humanoid Robot. IEEE Transactions on Systems, Man and Cybernetics, Part B, Special issue on robot learning by observation, demonstration and imitation, 37:2, 286-298.